

Alternativas para compartir conocimiento en Sistemas Multi-agente utilizando el modelo Linda

Luciano H. Tamargo

Marcelo A. Falappa

Alejandro J. García

Laboratorio de Investigación y Desarrollo de Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Email: {lt,mfalappa,ajg}@cs.uns.edu.ar

RESUMEN

Esta línea de investigación tiene como objetivo analizar el intercambio de conocimiento en sistemas multi-agente utilizando áreas de conocimiento compartidas. El proyecto involucra la extensión del modelo Linda, agregando nuevas características a nivel de modelo. En este trabajo se plantea la idea de espacios de tuplas distribuidos, brindando la posibilidad de coexistir diferentes espacios de tuplas con características distintas, residiendo en áreas de memorias remotas.

Palabras claves: Sistemas Multi-Agentes, Agentes, Ambiente Distribuido, Linda, Conocimiento compartido.

1. INTRODUCCIÓN

En un sistema multi-agente los agentes deben poder intercambiar información, ya sea en un entorno colaborativo o basado en negociación. Este intercambio puede hacerse por medio de pasaje de mensajes o utilizando áreas de conocimiento compartidas. Este trabajo tiene como objetivo analizar el intercambio de conocimiento utilizando áreas de conocimiento compartidas. La importancia de esta investigación radica en que, disponer de un modelo para compartir conocimiento, es un aspecto a resolver en sistemas multi-agente con agentes deliberativos. Por lo tanto, sería interesante contar con un modelo para mantener las bases de conocimiento que permitiera abstraerse sin preocuparse de la implementación de las mismas.

El modelo Linda [1, 4, 3, 6, 7, 2, 5], es un modelo de memoria compartida originalmente definido para proveer comunicación y sincronización en programas con paralelismo. Utiliza un espacio de memoria compartida llamado “*espacio de tuplas*” (ET), sobre el cual propone seis operaciones (*out*,

in, *rd*, *eval*, *inp* y *rdp*). Agregando estas operaciones del ET a un lenguaje base se genera un dialecto de programación paralela.

Los ET son una abstracción a partir de la cual los procesos cooperan y se comunican. Estos pueden verse como áreas de memoria compartidas asociativas referenciadas que no requieren hardware subyacente para el área de memoria física donde residen.

Linda provee cuatro operaciones básicas de espacio de tuplas, *out*, *in*, *rd* y *eval*, y dos variantes adicionales *inp* y *rdp*. La operación **out(t)** causa que la tupla *t* sea agregada al ET; **in(s)** causa que alguna tupla *t* que concuerde con la tupla-molde *s* sea retirada del ET; **rd(s)** es igual a *in(s)*, excepto que la tupla que coincide no es retirada y permanece en el ET. Las versiones predicativas de *in* y *rd*, **inp** y **rdp** respectivamente, intentan localizar una tupla que coincida, y retornan 0 si la búsqueda falla, en caso contrario retornan 1. La operación **eval(t)** es igual a *out(t)*, excepto que *t* es evaluado después (en paralelo) en vez de antes de ser ingresado en el ET. Para mayor información acerca de los detalles del modelo Linda y sus operaciones, consultar [1, 4, 3, 6, 7, 2, 5].

El artículo está organizado como sigue: en la sección 2 se plantea un ejemplo dentro del dominio de búsqueda colaborativa con el cual se introducirán los principales conceptos a desarrollar en este trabajo, en la sección 3 se hace mención a diferentes casos para vincular agentes con ETs, en la sección 4 se analizan distintas alternativas para asociar ETs con áreas de memoria, y en la sección 5 se plantea la conclusión y el trabajo a futuro.

2. EJEMPLO DE APLICACIÓN

Supongamos que se cuenta con cinco agentes que son robots (*Ag1*,...,*Ag5*) que se desplazan a través de un laberinto, ingresando por una puerta de entrada y buscando la única puerta de salida. Dichos agentes forman dos equipos (equipo 1: *Ag1*

Financiado parcialmente por Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) PIP 5050, y por la Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096)

y $Ag2$; y equipo 2: $Ag3$, $Ag4$ y $Ag5$) que compiten entre sí por encontrar la salida antes que su adversario lo haga. Ni bien los robots ingresan al laberinto, su conocimiento deberá ser sometido a subsecuentes procesos de actualización, incorporando creencias tales como: “la habitación de entrada tiene tres salidas posibles”, “no se puede girar a izquierda ni a derecha”, o cualquier información de este estilo representada en un lenguaje determinado. Los robots irán adquiriendo conocimiento para buscar la salida del laberinto. El mundo donde se desenvuelven los agentes puede cambiar ya sea por evolución propia, por acciones de los agentes en cuestión, o por acción de otros agentes. Uno de los casos en el cual se producirían cambios en el mundo es el siguiente: asumamos que el laberinto contiene puertas que una vez accionadas pueden quedar selladas o clausuradas. En ese caso, si un agente $Ag1$ clausura una puerta por acción propia y el agente $Ag2$ vuelve a pasar por la habitación que da contra esa puerta, notará un cambio en el mundo y deberá actualizar su conocimiento. \square

3. DIFERENTES CASOS PARA VINCULAR AGENTES CON ESPACIOS DE TUPLAS

Como se ha observado, el modelo Linda provee un único área de memoria que es compartida por todos los agentes que intervienen en el sistema multi-agente en cuestión. A continuación, se plantean diferentes casos que promueven una extensión al modelo mencionado, permitiendo la existencia de más de un ET en el sistema:

Caso 1: Un extremo opuesto al que propone Linda, sería mantener un ET por agente. Esto resulta útil en aquellas situaciones donde se requiere que los agentes puedan disponer de una base de conocimiento propia y exclusiva. En este caso, si los agentes desean comunicarse deberán hacerlo por medio del pasaje de mensajes.

Caso 2: Otro caso consistiría en disponer de un espacio de tuplas compartido por todos los agentes, y además proveer primitivas para que pares de agentes puedan crear espacios de tuplas “privados” (con el consentimiento de ambos), y así lograr que los agentes puedan compartir conocimiento sólo con aquellos agentes que deseen. Éste, es un caso intermedio entre el caso anterior y el propuesto por Linda (un único ET compartido por todos). Esta aproximación brinda una gran flexibilidad al modelo. Podría suceder que el espacio compartido por todos no existiera y que los agentes solo se comuniquen por medio de espacios de tuplas privados creados por ellos. De todas formas, es interesante mantener siempre un espacio de tuplas compartido por todos, para que

los agentes que intervienen en un sistema multi-agente se puedan conocer y llegar a un acuerdo para la creación de espacios de tuplas “privados”.

Caso 3: En este caso se plantean los espacios de tuplas “de grupos” que resultan de una extensión del caso anterior. Estos pueden ser accedidos por determinados agentes que estén relacionados de alguna manera. Por ejemplo, la relación puede estar determinada de la siguiente forma: todos los agentes del grupo trabajan en la misma tarea. En base a esta relación, los agentes que trabajan sobre el mismo objetivo comparten conocimiento, permitiendo así que la solución de un problema pueda ser dividida, para que cada agente ataque una parte y pueda contribuir a la solución global poniendo sus resultados en el espacio de tuplas.

En base a los casos mencionados aquí se pueden derivar varias combinaciones, y así obtener esquemas con diferentes características para lograr cubrir un espectro aún mayor de necesidades a la hora de desarrollar un sistema multi-agente.

Ejemplo: En el contexto del ejemplo del laberinto, explicado en la sección 2, es interesante contar con diferentes espacios de tuplas. El diseño del problema se encuentra expresado gráficamente en la figura 1, donde un rectángulo punteado representa a un espacio de tuplas, un círculo representa a un agente y una línea que conecta un círculo con un rectángulo punteado representa un vínculo entre un agente y un espacio de tuplas haciendo referencia a que dicho agente puede acceder a dicho espacio. Esta representación gráfica se mantendrá a lo largo de todas las figuras en este trabajo.

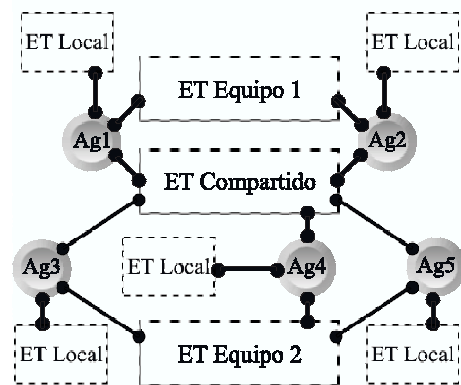


Figura 1: Diseño de los espacios de tuplas del ejemplo del laberinto

A continuación se explican cada uno de los espacios de tuplas que se encuentran representados en la figura 1:

► Un espacio de tuplas compartido por todos los agentes (ET Compartido): Este serviría para que

algún agente informe a los demás que ha encontrado la salida, de esta manera se logra que el resto de los agentes cesen sus búsquedas.

► Un espacio de tuplas privado compartido entre los agentes *Ag1* y *Ag2* (ET Equipo 1): Aquí se almacenará toda información relevante para el equipo 1. Los datos en este espacio serán utilizados por los integrantes del grupo para poder desarrollar algoritmos eficientes en busca de la solución. Por ejemplo, cada miembro del equipo podría almacenar los estados de las puertas de las habitaciones que ya visitó, con las respectivas coordenadas de la habitación. Esto es, podría almacenar en el espacio algo como “habitación 23,10; puerta 2; cerrada”. Con esta información los agentes del equipo saben que puertas están clausuradas y cuales no. Esto les brinda la posibilidad de planificar la solución sin tener en cuenta ciertas puertas, y de distribuir el trabajo de exploración logrando una labor colaborativa.

► Un espacio de tuplas grupal compartido entre los agentes *Ag3*, *Ag4* y *Ag5* (ET Equipo 2): cumple la misma función que el espacio privado explicado con anterioridad, pero es explotado por el equipo 2.

► Un espacio de tuplas local para cada agente (ET Local): Aquí cada agente almacenará todo lo que perciba, sin filtros. Esto último quiere decir que un agente, basándose en un “algoritmo de relevancia” de información, puede replicar los datos en el espacio que comparte con el resto del equipo, luego de agregarlos al espacio de tuplas local. Este “algoritmo de relevancia” sería una especie de filtro de información que selecciona las tuplas que son necesarias compartir con el grupo. Un ejemplo de lo que un agente puede guardar en forma local y no considera que sea necesario compartir con los demás miembros, es el camino que lleva recorrido, para no volver a repetirlo. □

4. ALTERNATIVAS PARA ASOCIAR ESPACIOS DE TUPLAS CON ÁREAS DE MEMORIA

Como un ET es un concepto abstracto, el mismo puede residir en diferentes áreas de memorias, o sólo en una. Asimismo un área de memoria puede albergar espacios de tuplas diferentes. Se puede notar también, que un ET puede ser compartido o no. En este contexto se podría pensar en diferentes organizaciones de los ET:

- T1- ET *compartido centralizado*
- T2- ET *compartido distribuido*
- T3- ET *no compartido centralizado*
- T4- ET *no compartido distribuido*

Por lo tanto, en lugar de proveer un único área de memoria compartida y centralizada (como lo sugiere Linda), se podría presentar en un mismo sistema multi-agente, varios espacios de tuplas con diferentes características. Debajo se apuntarán 6 alternativas que se vislumbran de las 4 organizaciones mencionadas arriba:

Alternativa 1: La primer alternativa consiste de disponer de un único espacio de tuplas compartido por todos y centralizado. Esta alternativa es mostrada gráficamente en la figura 2, donde un rectángulo de línea continua representa un área de memoria residiendo físicamente en un nodo. Notar que cuando un rectángulo punteado (ET) se superpone con un rectángulo de línea llena (área de memoria), significa que el ET está residiendo en esa área de memoria. Esta representación gráfica se mantendrá a lo largo de todas las figuras en este trabajo. Es importante destacar que si un agente está vinculado a un ET no necesariamente está ejecutándose en el mismo nodo.

Con esta alternativa puede representarse la organización **T1**, la cual es el esquema sugerido por el modelo Linda y tiene como ventaja su simpleza en cuanto a compartir conocimiento. Esto se debe a que, cuando un agente desea comunicarse con otro, sólo tiene que poner una tupla en el espacio para que el receptor la retire. Tiene como desventaja la inseguridad de la información. Esto último se debe a que todos los agentes pueden acceder a toda la información que se encuentra en el área de memoria compartida. Por ejemplo, teniendo en cuenta el esquema propuesto en la figura 2, si el agente *Ag1* desea enviar información al agente *Ag4*, pondrá una tupla en el espacio de tuplas (mediante una operación *out(t)*) con la información que enviará. Nada impide que el agente *Ag2* realice un *in(t)* antes que el destinatario, y de esta manera quite la tupla del espacio. Esto puede solucionarse incorporando un mecanismo de seguridad, como por ejemplo, agregando permisos de accesos. □

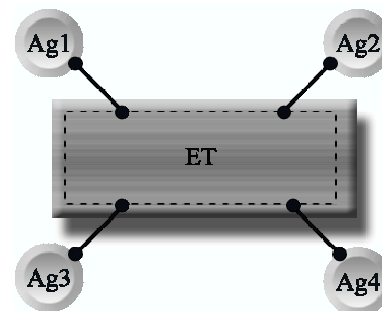


Figura 2: Espacio de tuplas compartido centralizado

Alternativa 2: La segunda alternativa es mostrada en la figura 3 donde, un único espacio de tuplas está residiendo en áreas de memorias remotamente distribuidas, con una filosofía de replicación de tuplas en cada nodo donde reside el espacio. En esta figura, cada rectángulo de línea llena representa las distintas áreas de memorias ubicadas físicamente en diferentes nodos, en las cuales reside el ET que está representado por el recuadro punteado que los contiene.

Un esquema con estas características representa la organización **T2**. En este caso ante una actualización, las acciones a tomar por la misma, deben ejecutarse en todos los nodos donde está residiendo el espacio de tuplas en cuestión. Esto es, ante una operación $out(t)$ en la figura 3, la tupla t debe ser agregada en todos los nodos.

Esta alternativa es deseada, cuando se requiere un sistema robusto, ya que la replicación de los datos provoca que ante la caída de un nodo no se pierda información. Aquí la comunicación se realiza mediante pasaje de mensajes, lo cual es una desventaja debido a que se debe conocer al receptor. □

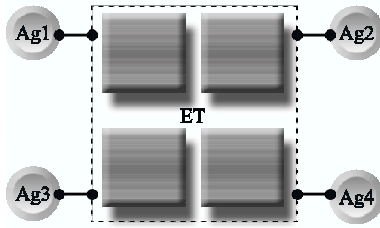


Figura 3: Espacio de tuplas compartido distribuido con replicación de datos

Alternativa 3: Una derivación del caso anterior puede verse en la figura 4 donde, un único espacio de tuplas está residiendo en áreas de memorias remotamente distribuidas y, a diferencia del caso anterior, aquí no se replican los datos en los distintos nodos, sino que en cada uno de ellos reside una región del espacio de tuplas a representar. Esto es, en cada nodo se podrá acceder a información que es local al mismo, y que es un subconjunto de la información perteneciente al espacio de tuplas en general. El total de la información contenida en el espacio de tuplas resulta de la unión de toda la información contenida en los diferentes nodos donde se alberga dicho espacio.

En la figura 4 se puede notar que se hace referencia a un único ET particionado, en donde cada partición del mismo se encuentra residiendo en diferentes áreas de memorias dispuestas remotamente.

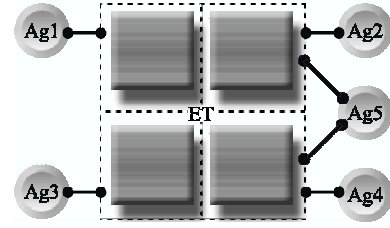


Figura 4: Espacio de tuplas compartido distribuido sin replicación de datos (particionado)

Un esquema con esta característica representa la organización **T2** y resulta de utilidad cuando el tamaño del espacio de tuplas es muy grande y se necesitan varias áreas de memoria para albergar al mismo. □

Alternativa 4: En la figura 5 se puede observar un esquema de sistema multi-agente que consiste de un espacio de tuplas por agente, con una disposición distribuida del sistema. Con esta alternativa se puede obtener un sistema multi-agente cuyos espacios de tuplas residan en nodos remotamente distantes. Los espacios de tuplas no son compartidos y la comunicación aquí, se realiza por medio de pasaje de mensajes. Con esta alternativa se representa la organización **T4**.

Con este tipo de implementaciones se vuelve a caer en las desventajas de la comunicación por medio de pasaje de mensajes, esto es, cada vez que un agente desee comunicarse con otro deberá conocer la existencia del receptor (esto no sucede en Linda). Sin embargo, se obtienen las ventajas de los sistemas distribuidos, y la información emitida por un agente no es alterada por alguna entidad indeseada.

Esta alternativa surge de la necesidad de proveer a cada agente de un área de memoria que sea inaccesible al resto de los agentes. Esto permite que cada agente mantenga información privada que es solo de interés local. □

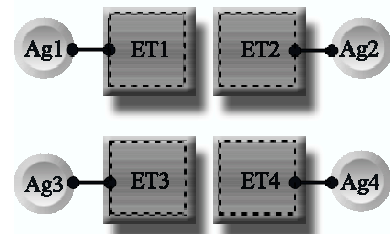


Figura 5: Espacios de tuplas no compartidos totalmente distribuidos

Alternativa 5: Una de las últimas alternativas puede verse representada gráficamente en la figu-

ra 6. Ésta consiste de varios ETs residiendo en un mismo área de memoria. Un esquema de estas características representa la alternativa **T3**, en la cual se mantienen los ETs centralizados pero éstos no son compartidos. El hecho de tener los ETs con esta disposición permite que varios agentes mantengan en un mismo nodo sus respectivos ETs, sin la necesidad de compartirlos.

En el caso de que un sistema multi-agente requiera, para su ejecución, varios ETs que no sean compartidos, y no se cuente con áreas de memorias remotas como para albergar cada ET (falta de recursos), esta alternativa brinda la posibilidad de centralizar los ETs en un mismo área de memoria permitiendo así coexistir en un mismo nodo varios ETs consumiendo una menor cantidad de recursos.

Por otra parte, esta alternativa tiene como desventaja, que si el nodo donde se centralizan los ETs se cae, se pierde toda la información, lo cual no sucede si dichos espacios se encuentran distribuidos. Esto evidencia lo poco robusta que resulta esta alternativa. □

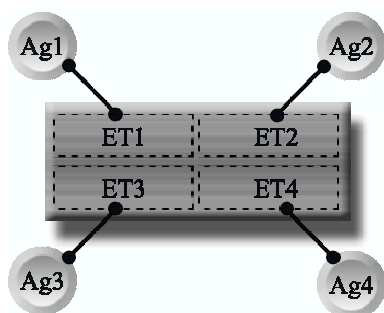


Figura 6: Esquema centralizado de espacios de tuplas no compartidos

Alternativa 6: Una variación de la alternativa anterior consiste en poseer varios ETs compartidos residiendo en un solo área de memoria. Su representación gráfica es similar a la figura 6, la única diferencia es que pueden existir más de un agente vinculado a cada ET, reflejando de esta manera un comportamiento compartido de los ET.

Como se puede observar, esta alternativa posee las mismas ventajas (ideal ante la falta de recursos) y desventajas (esquema poco robusto) que la anterior. □

En base a los casos mencionados en la sección 3 y a estas alternativas, se pueden derivar varias combinaciones. Esto es, se pueden obtener ET, ya sean “privados”, “grupales”, etc., que pueden ser implementados persiguiendo cualquier alternativa mencionada en esta sección.

5. CONCLUSIÓN

En este trabajo se analizó el intercambio de conocimiento en sistemas multi-agente utilizando áreas de conocimiento compartidas. Esto involucró una extensión del modelo Linda, en la que se agregó una nueva característica a nivel de modelo. En la misma se planteó la idea de espacios de tuplas (ETs) distribuidos, brindando la posibilidad de coexistir diferentes ETs con características distintas, residiendo en áreas de memorias remotas. Este planteo se hizo explícito en las 6 alternativas nombradas en la sección 4. Cada una de estas tiene sus ventajas y desventajas. En resumen, la alternativa 1 logra un ambiente simple para compartir conocimiento, pero inseguro; las alternativas 2, 3 y 4 tienen como desventaja la comunicación por medio de pasaje de mensaje pero, en la alternativa 2 se mantiene robustés, en la alternativa 3 se brinda la posibilidad de distribuir ETs muy grandes en más de un área de memoria, y en la alternativa 4 se provee a cada agente de un área de memoria local; las dos últimas alternativas (5 y 6) permiten que se centralicen varios ETs ante la falta de áreas de memorias, pero son poco robustas.

Referencias

- [1] N. Carriero and D. Gelernter. Linda in context. *Communications of the ACM*, 32(4):444–458, 1989.
- [2] N. Carriero and D. Gelernter. Capitulo 3: Linda. In *How to write parallel programs*, 1992.
- [3] Paolo Ciancarini. Coordination Languages as Software Integrators of Multiagent architectures. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 21(4):314–335, 1995.
- [4] Paolo Ciancarini. Coordinating Multiagent Applications on the WWW: A Reference Architecture. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 24(5):362–374, May 1998.
- [5] Marcelo A. Falappa and Alejandro J. García. Extensión del modelo linda para representar la dinámica de conocimiento en sistemas multi-agente. In *Proceedings of X Argentine Congress of Computer Science*, 2005.
- [6] Kurt Schelfhout and Tom Holvoet. An Environment for Coordination of Situated Multi-Agent Systems. 2004.
- [7] Danny Weyns, H. Van Dyke Parunak, Fabien Michel, Tom Holvoet, and Jacques Ferber. Environments for Multiagent Systems State-of-the-Art and Research Challenges. 2004.